تطوير نظام التعرف على الارقام المكتوبة بخط اليد باستخدام تقنيات الشبكة العصبية ANNوCNN

سامية موسى [1] , محمد صالح[2]

1- جامعة بنغازي, كلية تقنية المعلومات, قسم علم الحاسوب, فرع المرج, samiacomp.sc@gmail.com
2- جامعة درنة, كلية العلوم, قسم الحاسوب, فرع القبة , mohammed.saad@omu.edu.ly

الملخـص

أصبحت مشكلة التعرف على الأرقام المكتوبة بخط اليد واحدة من أهم مشاكل التعلم الآلي وتطبيقات الرؤية الحاسوبية ,وقد تم استخدام العديد من تقنيات التعلم الآلي لحل مشكلة التعرف على الأرقام المكتوبة بخط اليد. تم في هذا البحث عمل دراسة مقارنة لخوارزميتين من خوارزميات التعلم الآلي وهي Artificial Neural Network (ANN) و Convolutional Neural Network (CNN) ,وتم استخدام مجموعة بيانات خاصة بالأرقام المكتوبة بخط اليد, MNIST(Modified National Institute of Standards and Technology database). تتم المقارنة بناء على الدقة والأداء مع طبقات Batch Normalization و Dropout ,وبدونها لملاحظة تحسن الأداء .تم التوصل لنتائج جيدة فقد تحصلت CNN على نسبة أعلى من ANN .

1- المقدمـة

إن حاسة النظر هبة وهبنا الله إياها ,بحيث يمكننا رؤية الأشياء المحيطة من حولنا والتعرف عليها بسهولة, ولكن حاسة النظر تعتبر عملية معقدة, حيث يتم إرسال إشارات إلى الدماغ والقشرة البصرية بالتحديد, حتى يتسنى لنا فهم ومعرفة ما رأيناه. مع التطور التقني الذي نشهده أهتم العلماء الباحثين والمطورين بدمج التكنولوجيا بعالمنا الواقعي, واحد من هذه المجالات هو الرؤية الحاسوبية(Computer Vision), خصوصا مع انتشار العديد من الصور والفيديو والكاميرا على الانترنت, وتتفوق الحواسيب على الإنسان في الدقة العالية التي تتميز بها في التقاط الصور ومعرفة الألوان, إلا أن الإنسان لديه القدرة على فهم ومعرفة ما يحيط به أو ما يراه, بينما الحاسوب يرى الصور عبارة عن Pixels وقيم عددية تحدد الألوان فقط.

هذا الأمر أصبح محط اهتمام العلماء, وهو جعل الحواسيب (الآلات) تفهم العالم الحقيقي المحيط بنا, وقد ظهرت عدة أبحاث وتم تطوير عدة نماذج استطاعت أن تتعرف وتصنف الأشياء. واحد من هذه المجالات التي يندرج تحت الرؤية الحاسوبية هو التصنيف(Classification) ,مثل التعرف على الوجه ,أو التعرف على الكلام ,أو التعرف على خط اليد. التعرف على خط اليد هو نوع من التعرف الضوئي على الحروف (OCR), ويقصد به قدرة الألة على التعرف وتفسير النصوص المكتوبة بخط اليد, ويمكن أن تتم عملية التعرف بالاتصال بالإنترنت(Online) أو من دون انترنت (Offline)وفي هذا البحث نهتم بالتعرف على الأرقام المكتوبة بخط اليد ,باستخدام مجموعة البيانات, Modified) National Institute of Standards and Technology database).

التعرف على الأرقام المكتوبة بخط اليد (Handwritten Digit Recognition) يتم عن طريق إدخال صورة مكونة من الرقم المراد التعرف عليه ,ثم يتم إخراج الرقم المفترض التعرف عليه .ساعدت هذه التقنية في حل العديد من المشاكل ,مثل التعرف على أرقام الشيكات المصرفية.

من الخوارزميات التي يمكن استخدامها وقد أظهرت مدى دقتها وفعاليتها في التعرف هي الشبكات العصبية الاصطناعية Artificial Neural Network (ANN), وهي نماذج تحاكي دماغ الانسان . تقليد للشبكة العصبية البيولوجية ولكن بمنظور أخر, بمعنى تشابه في طريقة الأداء ولكن بشكل مختلف و الخوارزمية المستخدمة الثانية تسمى الشبكات العصبية الالتفافية Convolutional Neural Network (CNN) وهذه الشبكات نوع من أنواع الشبكات العميقة Deep Neural Networks ,والتي أثبتت فعاليتها وأدائها خصوصا في أنها مختصة بمجال التصنيف الصور هذا الأمر جعل منها أكثر استخداما.

مع التطور السريع الذي نشهده ,والتوجه نحو العالم الرقمي .نلاحظ الاحتياج اكثر لتسهيل وتسريع العمل من ادخال وكشف وارسال سواء في المؤسسات الحكومية أو المدارس ,أو المصارف. يسعى دائما الى رقمنه المهام وأرشفتها لسهولة الوصول إليها, وهذا الأمر قد يكون متعبا ويحوي الكثير من الأخطاء بسبب الإدخال اليدوي لقاعدة البيانات. والهدف من هذا الورقة إنشاء شبكة عصبية اصطناعية, (ANN) وشبكة عصبية التفافية,(CNN) وتدريبها باستخدام مجموعة البيانات,(MNIST) للمقارنة بينهما. وكذلك استخدام طبقات (Batch Normalization) و (Dropout) ,لتجنب مشكلة (Overfitting) ثم مقارنة النتائج للتوصل لأفضل أداء لشبكة العصبية في التعرف على الأرقام المكتوبة بخط اليد, ولكن يعد تدريب الشبكات العميقة مهمة صعبة, تتضمن العديد من المشكلات التي يجب معالجتها ,فبالرغم من إمكاناتها الهائلة ,إلا أنها يمكن أن تكون بطيئة وعرضة لمشكلة فرط التجهيز(Overfitting).

هناك العديد من التقنيات لحل مشكلة Overfitting ,ومن هذه التقنيات هي الـ(Batch Normalization),وهي واحدة من التقنيات التي تستخدم بشكل شائع في التعلم العميق ,من ميزاتها أنها تحسن من سرعة الشبكات العصبية ,وتوفر التنظيم (Regularization),كما تمنع من فرط التجهيز.

2- الدراســات السابقــة

2-1 الدراسات السابقة للشبكة العصبية الاصطناعية(Artificial Neural Network)

زاد اهتمام الباحثين والأكاديميين ,في السنوات الأخيرة في العمل على زيادة كفاءة التعرف على

الأرقام المكتوبة بخط اليد ,وتحسين دقته .يحتاج التعرف على الأرقام المكتوبة بخط اليد الى مجموعة بيانات كبيرة نسبيا, ووقت تدريب طويل لتحسين الدقة وتقليل نسبة الخطأ. [1] يقترح نهج نموذج لشبكة عصبية اصطناعية تعمل بكفاءة وسرعة ,وتعمل بطريقة التدريب المتوازية على وحدة معالجة الرسومات GPU,للتقليل من وقت التدريب ,وتم مقارنة النهج المقترح المعتمد على خوارزمية PBMLP اعتمادا على وحدة معالجة الرسومات GPU مع النهج الحالي المعتمد على وحدة المعالجة المركزية CPU ,وتبين من المقارنة كفاءة المنهجية المقترحة حيث وصلت لدقة في تصنيف الارقام بنسب بين 100-97 بالمئة لكل رقم, بينما منهجية التي تعتمد على CPU كانت نسبها تصل بين 100-87 بالمئة. وبالمثل قد كان التعرف على الأرقام المكتوبة بخط اليد مجالا رئيسيا للبحث في التعرف البصري (OCR) . الهدف منه تنفيذ شبكة عصبية (MLP) ,للتعرف والتنبؤ بالأرقام من 0 إلى 9, باستخدام MNIST ,وبالاعتماد على خوارزمية الانتشار الخلفي (back propagation) في مرحلة التدريب ,واختبارها باستخدام خوارزمية التغذية الأمامية (feed forward) ,وقد وصلت دقة النظام المقترح إلى 99.32 بالمئة [2].

تم في هذه الورقة عمل دراسة استقصائية (Survey) ,حول منهجيات الشبكات العصبية باختلافها للتعرف على الأرقام المكتوبة بخط اليد, تم فيها مقارنة بين ثلاث خوارزميات ,من بينها خوارزمية الانتشار الخلفي التي قدمت أداء أفضل من السرعة والفعالية مقارنة بالباقي, كما أن يعطي تقارب سريع لــlocal minima القيم الدنيا في حال وجود خطأ [3].

التعرف على الأرقام والحروف هي مهارة مطلوبة في صناعة التطبيقات الحالية ,مثل: معالجة الاشارات, والتعرف على العملات, والتعرف على رقم المنزل. تركز الورقة على أداء الشبكة وفقا للمعالجة المسبقة للبيانات مع التدرج الرمادي ,بالإضافة إلى استخراج الميزات ,تم استخدام ANN للتعرف على الأرقام باستخدام Octave, والتنظيم لتجنب مشكلة Overfitting ,وقد تبين أن الشبكة تنفذ في وقت أقل مع دقة بنسبة97.5 بالمئة[4].

التعرف على الخط(HWR) ,هو قدرة الحاسوب على استقبال وتفسير مدخلات مكتوبة بخط اليد, من مصادر ,مثل :المستندات الورقية ,والصور, وشاشات اللمس . تمت المقارنة بين ثلاث خوارزميات وهي: (SVM) و (KNN) و (ANN) ,وقد ركزت الدراسة في المقارنة على الدقة, وتبين أن أفضل دقة تم الحصول عليها هي 99.26 بالمئة للخوارزمية KNN ,وتبين كذلك أن أداء كل من KNN و SVM أدائهما أفضل بكثير من MLP [5].

[6] دراسة مقارنة لثلاثة خوارزميات وهي (Naïve Bayes) و (MLP) و(K_star) ,وقد تمت المقارنة بناءا على دقة الأداء, بحيث تركز الدراسة على أي الخوارزميات تعطي دقة أفضل مقابل عدد ميزات أقل(Features) ,وقد أعطت خوارزمية K_star دقة أعلى, وتعتبر دقة مقبولة بنسبة 82 بالمئة.

2.2 - الدراسات السابقة للشبكة الالتفافية(Convolutional Neural Network)

في السنوات الأخيرة الماضية أظهرت أساليب التعلم العميق ,وبالأخص الشبكات العصبية التلافيفية (CNNs) ,أظهرت دقة ممتازة في الكثير من مشاكل تصنيف الأنماط(Pattern Classification). تتضمن دراسة حالة للشبكات الالتفافية(Convolutional Networks) بهيكليات مختلفة ,وقد ركزت هذه الدراسة على إعطاء لمحة عن المعالجة المسبقة للبيانات (Pre-Processing) ,و زيادة حجم بيانات التدريب (augmentation data), وبينت أهمية المعالجة المسبقة, وكيف أن لها تأثير على مدى أداء الشبكة ,وكيف أن زيادة حجم البيانات يساعد على تكرار البيانات على الشبكة ,مما يساعدها على التعلم, وتم في هذه الدراسة المقارنة بين أنواع مختلفة من الشبكة الالتفافية ,وأظهرت النتائج أداء جيد يصل الى 99 بالمئة لكل شبكة على اختلاف هيكلية كل واحدة منها [7]. لقد أخذ التعلم العميق منحى ,وتغير جذريا وبذلك أصبح أكثر ذكاء ,بفضل ظهور الشبكات العصبية. [8] وتقوم الدراسة على مقارنة ثلاث أنواع من الشبكات من نوع الشبكة العميقة ,وتركز على تحديد أفضل خوارزمية بناءا على عدة عوامل من أهمها الدقة والأداء ,وكذلك آلية التنفيذ .تم استخدام مجموعة بيانات لتقييم الخوارزميات ,من خلال النتائج تبين (DNN) أفضل خوارزمية ,من حيث الدقة والأداء ,بينما الخوارزميات الأخرى (CNN) و(DBN) متشابهة ومتقاربة في الدقة. [9] عمل مقترح الهدف منه تحسين لبنية الـ(CNN) مع ضبط دقيق للمعلمات ,مع استخدام مجموعة البيانات (MNIST) ,وركز العمل على تجنب المعالجة المسبقة للبيانات مع زيادة في عدد الطبقات والمقارنة بينها, وقد تبين من النتائج أن الشبكة ذات الأربع طبقات أدت إلى دقة أفضل ,وتصل إلى 99.76 بالمئة. وتبين أن الضبط الدقيق للمعلمات يساعد كثيرا في تحسن أداء الشبكة ,بالإضافة لخوارزمية التحسين (Adam) التي حققت الشبكة بجانبه دقة لـــ99.89 .

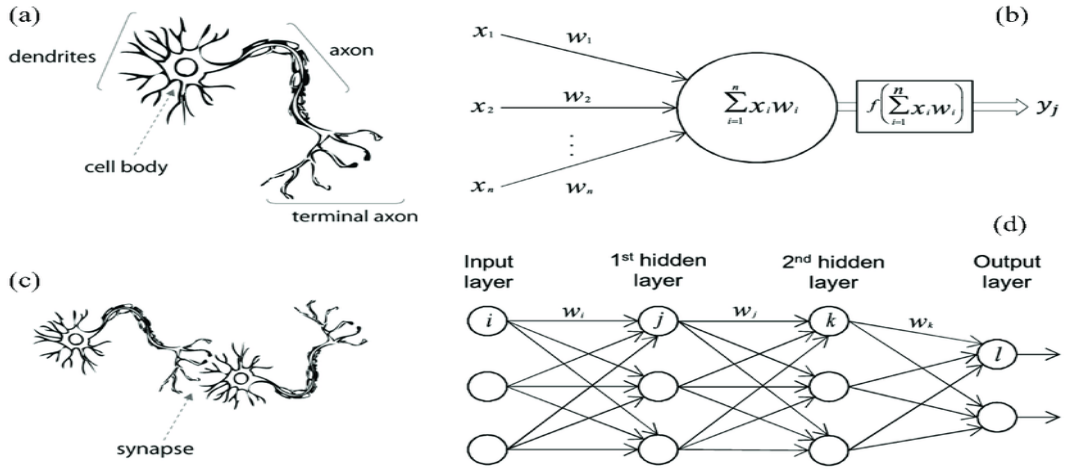
[10] تم في هذه الورقة تقديم تطبيق للتعرف على الأرقام المكتوبة بخط اليد,تم تطوير نموذج CNN يمكنه التعرف على الأرقام من الصور, بحيث يمكنه التنبؤ بدقة عالية ,تم التوصل لدقة 99.15 بالمئة, ويمكن لهذا النموذج من تحويل الارقام الى صيغة رقمية, والغرض الأساسي لهذا المقترح ,هو رقمنه كل شيء تلقائيا, تجنبا للأخطاء التي قد تحدث في حال إدخال البيانات الى قاعدة البيانات يدويا.

3- الأدوات والتقنيات المستخدمة

في هذا الباب سنتحدث عن الشبكات العصبية الاصطناعية والالتفافية العميقة ,وسنتطرق إلى الأدوات التي تم استخدامها لبناء النماذج وتدريبها, مع التحدث عن الطبقات والخوارزميات وكذلك التقنيات التي ساعدت في تطور النظام.

1.3 الشبكات العصبية الاصطناعية ANN

هي شبكة عصبية اصطناعية تحاكي آلية عمل الدماغ ,وتم استلهامها من الدماغ والأعصاب البشرية ,كما في الشكل التالي:



الشكل(1):يوضح الشبكة العصبية

نلاحظ في الشكل (1.3) ,الآتي :

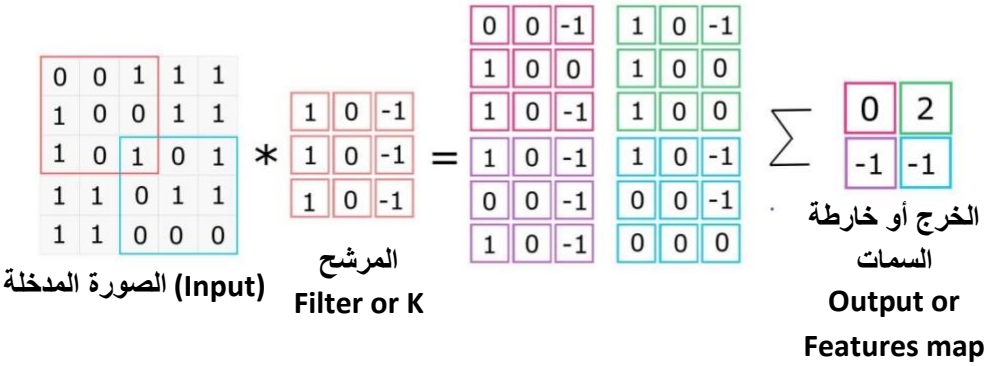
- إشارات الدخل Input(x1,x2,…xn) ,والتي تمثل إشارة المدخلات للخلية العصبية
- الأوزان Weight(w1,w2,….wn) , تشير الأوزان هنا الى الترابط ما بين العناصر
- تابع التنشيط أو التفعيل (Activation Function), وظيفته إخماد العصبون ,وجعل نطاقه بين 0و1
- الخرج Output(Yij),وهو الناتج النهائي للعمليات داخل الشبكة العصبية

تتكون الشبكة العصبية الاصطناعية(Artificial Neural Networks) ,واختصارها (ANNs) ,من ثلاث طبقات أساسية :طبقة الإدخال, والطبقة المخفية, وطبقة الإخراج. تتكون كل طبقة من عقد (nodes) أو كما تسمى ايضا خلايا عصبية(neurons) ,وكل عقدة تكون مترابطة مع العقدة في الطبقة التالية. يعتمد عدد العقد في طبقة الإدخال والإخراج على الخصائص والسمات الموجودة في مجموعة البيانات, أما بالنسبة للطبقات المخفية فيمكن زيادتها بحسب المشكلة المراد حلها, دون قيود.

2.3 الشبكات العصبية الالتفافية (Convolutional Neural Networks)

الشبكات العصبية الالتفافية, واختصارها (CNNs) ,تعتبر شبكات عصبية اصطناعية عميقة, يتم استخدامها في تصنيف الصور (Image recognition), وكذلك في التعرف على الأشياء (Object recognition). تتكون الــ(CNNs) ,من أربع طبقات أساسية ,وهي:

- طبقة الالتفاف(Convolute Layer) ,وهي الطبقة الأولى والرئيسية في الـــ(CNN) ,تستخدم في هذه الطبقة ما يسمى نواة (Kernel) أو مرشح(Filter) ,يتم عن طريقه مسح الصورة ,من ثم جمع البكسل (Pixel) ,واستخراج القيم النهائية كميزات(Features) ,تسمى خارطة ميزات (Features Map) ,الشكل التالي يوضح آلية الـــ(Convolute Layer) :

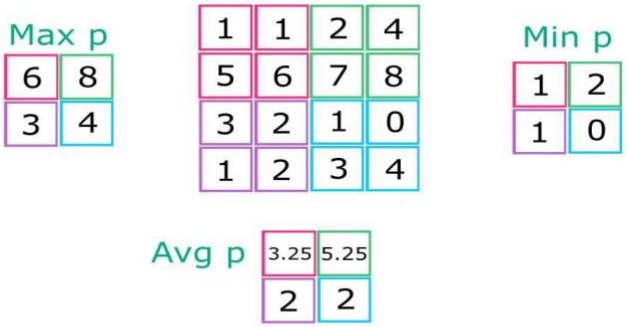


الشكل(2) : يوضح آلية عمل طبقة الالتفاف

- طبقة التجميع(Pooling Layer) ,غالبا ما تأتي هذه الطبقة بعد طبقة الالتفاف, وتقوم هذه الطبقة ,بتقليل أبعاد الصورة, وتتكون هذه الطبقة من عدة أنواع ,منها :

1. طبقة (Max pooling): تأخذ أعلى قيمة
2. طبقة (Average Pooling): تأخذ متوسط القيم
3. طبقة (Min Pooling) :تأخذ أقل قيمة

الشكل التالي يوضح آلية عمل طبقة التجميع :



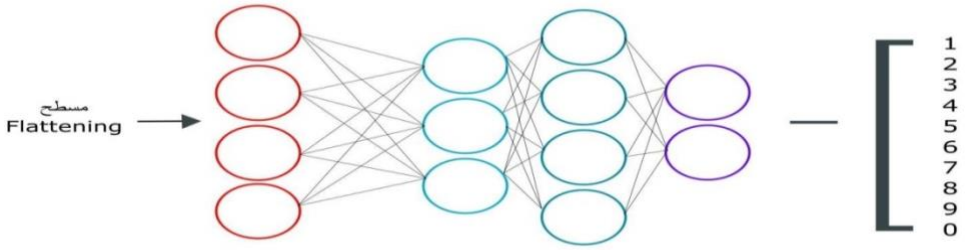
الشكل (3.3): يوضح آلية عمل طبقة التجميع

*نستخدم في هذا البحث الـ(Max pooling) ,لأنها أكثر شيوعا واستخداما ,وتفيد في تقليل الأبعاد أكثر.

- طبقة التسطيح (Flatten Layer),مهمة هذه الطبقة هي تسطيح البيانات من مصفوفة (Array) ,إلى متجه (Vector), وبذلك تصبح البيانات مسطحة كأنبوب.

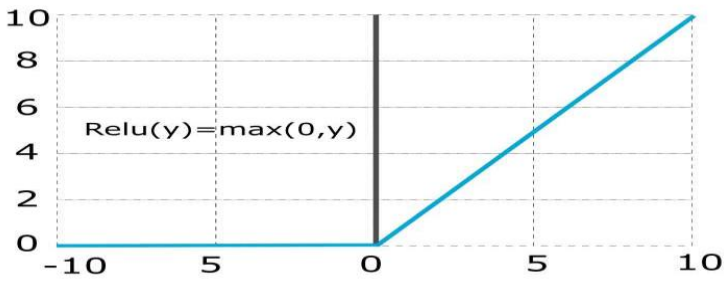
الشكل(3):يوضح آلية عمل طبقة التسطيح

- طبقة الاتصال بالكامل(Fully Connected) ,وهي الطبقة الأخيرة ,والتي يتم فيها ربط العصبونات ببعضها ,من أجل إخراج النتيجة النهائية.

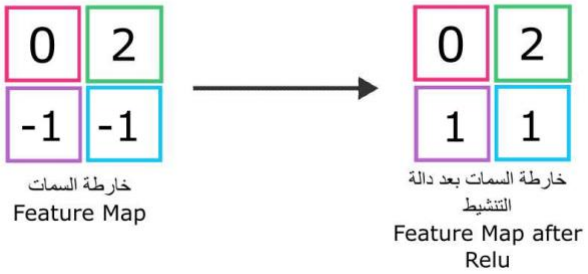


الشكل (4) يوضح الية عمل طبقة الاتصال بالكامل

- طبقة التنشيط نوع (Rectified Linear Unit) ,هذه الطبقة غالبا ما تستخدم مع طبقة الالتفاف ,وتسمى اختصارا (Relu) ,تقوم بتحويل المدخلات التي تم ترشيحها أو فلترتها ,إلى قيمة معينة ,ومن ثم إرسالها للطبقة التالية. نطاق هذه الطبقة هو (0,1), حيث تقوم هذه الطبقة بإخراج القيمة كما هي اذا كانت موجبة ,أما إذا كانت سالبة فالناتج يكون هنا صفر. تعتبر هذه الطبقة أو كما تسمى ايضا دالة أكثر استخدما في الطبقات العميقة ,وذلك لأنها تقلل من وقت التدريب. الشكل التالي يوضح دالة (Relu): الشكل(5):يوضح حساب دالة التنشيط Relu و الشكل(6) ,يوضح آلية عمل الدالة (Relu)

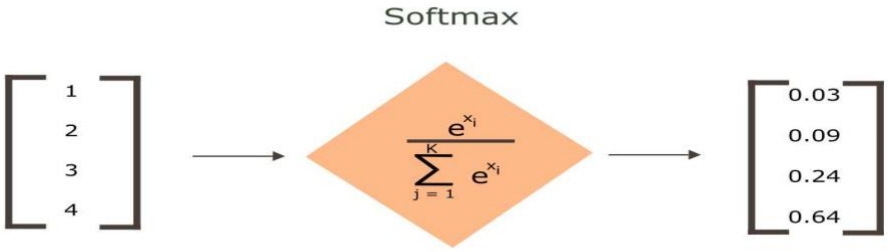


الشكل(5):يوضح حساب دالة التنشيط Relu



الشكل(6): دالة التنشيط Relu

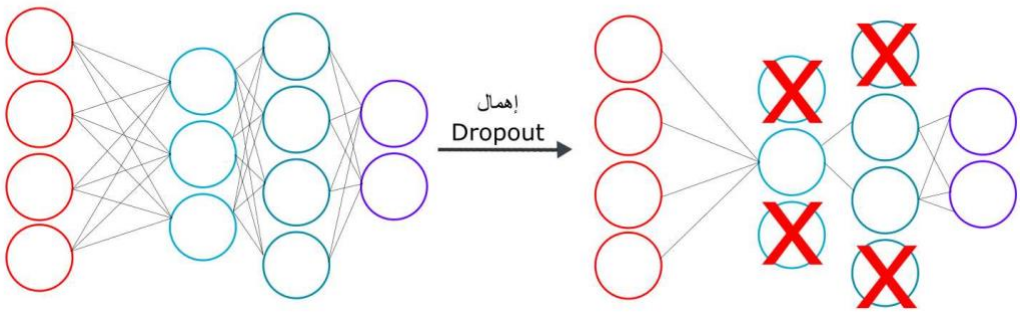
- دالة التنشيط (Softmax Function) ,تستخدم في الطبقة الأخيرة وفي التصنيف المتعدد ,تقوم هذه الدالة بتحويل متجه مكون من قيم حقيقية ,تمثل بـK ,الى متجه بقيم حقيقية ,ومجموعها يساوي 1,أي تقوم بتحويل أي قيمة سواء سالبة ,أو موجبة ,أو صفرية ,الى قيمة محصورة بين(0,1),ويتم اعتبار هذه القيم كاحتمالات(Probabilities) .الشكل (7), يوضح آلية عمل الــ(Softmax) :



الشكل(7):دالة التنشيط Softmax

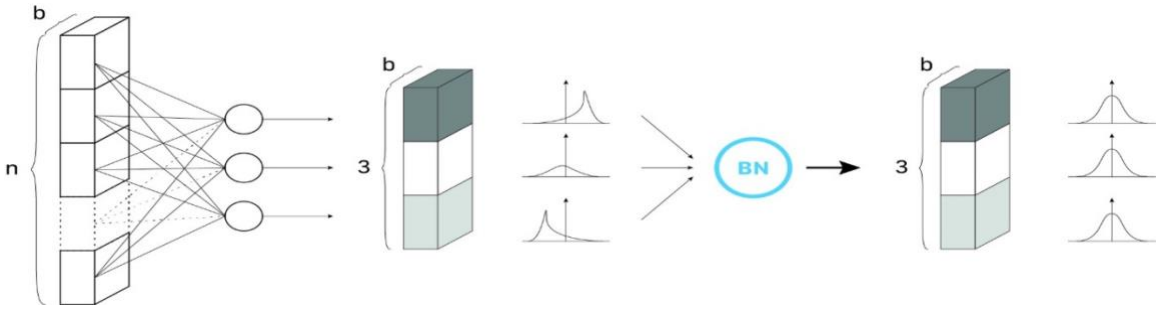
- طبقة الإهمال (Dropout Layer) , طبقة تستخدم أثناء تدريب الشبكة ,وذلك لتجنب مشكلة تعتبر شائعة في الشبكات العصبية ,تسمى(Overfitting) ,وآلية عمل طبقة الإهمال ,تتمثل في إيقاف أو تعطيل بعض العصبونات

في الطبقات المخفية (Hidden Layers), أثناء التدريب (Training), ويتم اختيار العصبونات بشكل عشوائي. توضح في الشكل (8):



الشكل(8): طبقة الإهمال

- طبقة التطبيع (Batch Normalization), هي عبارة عن طبقة ,يمكن إدراجها ضمن طبقات الشبكة العصبية, وظيفة هذه الطبقة هي تسوية بيانات الدخل ,وجعلها في نطاق بين (0,1),بحيث يجعلها متمركزة في الجزء الخطي لدالة التنشيط .تجعل هذه الطبقة الشبكة أسرع وقادرة على التعميم بشكل أكبر, الشكل (9) يوضح الــ Batch Normalization .



الشكل(9):يوضح طبقة الــ(Batch Normalization)

3.3 – قاعدة البيانات المستخدمة هي Modified National Institute of Standards and Technology database (MNIST)

هي مجموعة بيانات(Data Set) ,لأرقام مكتوبة بخط اليد ,وتم مقايستها لحجم معين و هو 28*28 بكسل. يتم استخدامها لتدريب واختبار النماذج المختلفة (Models), في معالجة الصور. تتكون MNIST من 70,000 صورة, مقسمة إلى 60,000 للتدريب(Training) و10,000 للاختبار(Testing) .

الشكل(10): مجموعة البيانات الخاصة بالأرقام المكتوبة بخط اليد

4.3 منصة Google Colab

هو منصة (Platform) برمجة وتطوير مجانية ,لغرض إتاحة التعلم للجميع ,بمواصفات حواسيب عالية المستوى ,كما أن يوفر لغات برمجة ,اهمها :C++,Python3, ولغة R.توفر كذلك مكتبات ذكاء اصطناعي مثل , Tensor Flow, وKeras, دون الحاجة لتحميلها وتثبيتها. تحتاج المنصة الى اتصال بالإنترنت فقط وتسجيل حساب, ويتم تخزين المشاريع على (Google Drive).

الشكل(11) : منصة جوجل كولاب

5.3 لغة البرمجة (Python)

هي لغة برمجة عالية المستوى, مترجمة (Interpreter) تفاعلية كائنية ,تعتبر سهلة التعلم كبداية, ذلك لأنه تستخدم كلمات إنجليزي بسيطة وغير معقدة ,بعكس اللغات التي تستخدم رموز, كما أن صياغتها واضحة ,ويسهل التعامل معها ,بخلاف اللغات الأخرى.

6.3 مكتبة (Keras)

هي مكتبة عالية المستوى ,مكتوبة بلغة بايثون, وتعتمد في أدائها على مكتبة (Theano),وخاصة بإنشاء شبكات عصبية اصطناعية, وتتميز هذه المكتبة بسرعة أدائها أثناء التدريب.

7.3 مكتبة (Numpy)

هي من مكتبات بايثون, وتختص بالمصفوفات ,تتميز بسرعة تنفيذ العمليات الحسابية, وذلك لأنها تحتوي على العديد من الدوال (Function), والطرق(method), التي تساعدنا على تنفيذ العمليات الحسابية على المصفوفات بسرعة.

8.3 مكتبة (Open CV)

مكتبة رؤية بالحاسوب مفتوحة المصدر, وهي تعتبر مكتبة من مكتبات التعلم الآلي لتطبيقات الرؤية بالحاسوب. تحتوي هذه المكتبة على عدة دوال تتعامل مع الصور ,حيث تقوم الدوال بتحديد عناصر محددة في الصور ,كالخطوط والأشكال ,والوجوه....,الخ.

9.3 مكتبة (Matplotlib)

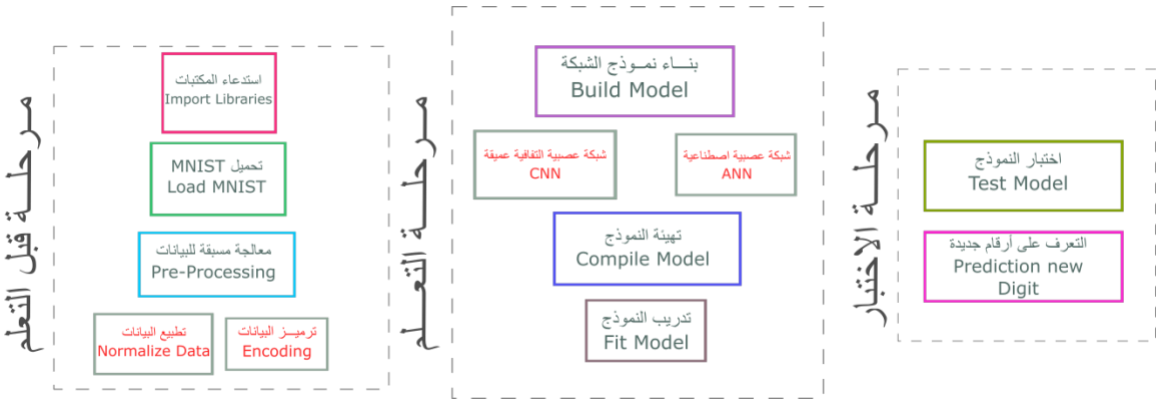
مكتبة تختص بإنشاء رسومات بيانية لعرض البيانات, وتفيد في مرحلة تحليل البيانات.

4 المنهجية

يتناول هذا الباب شرح مفصل لمنهجية البحث التي توضح جميع المراحل التي مر بها المشروع حتى اكتماله.

1.4.منهجية البحث

توضح منهجية البحث المراحل التي مر بها المشروع من مرحلة إعداد البيانات(معالجة مسبقة), الى مرحلة الاختبار والتعرف .الشكل(12)يوضح مراحل المشروع.



الشكل(12):يوضح مراحل المشروع

1.1.4- مرحلة قبل التعلم (مرحلة إعداد البيانات)

وهي المرحلة التي يتم فيها الإعداد الأولي للبيانات ,بالإضافة الى استدعاء البيانات ,وتحميل MNIST .

- **استدعاء المكتبات:**

الشكل (13)يوضح استدعاء المكتبات لـــ(ANN) :

```
#import libraries
import keras
from keras.models import Sequential,Model
from keras.layers import Dense, Flatten, Dropout, BatchNormalization
from keras.datasets import mnist
from keras.utils.np_utils import to_categorical
from keras.utils.vis_utils import model_to_dot
import matplotlib.pyplot as plt
import numpy as np
```

الشكل(14) استدعاء مكتبات لـANN

والشكل(15)يوضح استدعاء مكتبات لـCNN

```
#import libraries
import keras
from keras.models import Sequential
from keras.layers import Conv2D , MaxPool2D
from keras.layers import Dense, Flatten ,Dropout ,BatchNormalization
from keras.datasets import mnist
from keras.utils.np_utils import to_categorical
from keras.utils.vis_utils import model_to_dot
import matplotlib.pyplot as plt
import numpy as np
```

الشكل(15)استدعاء مكتبات لـCNN

لا يوجد فرق في المكتبات فقط أستخدمنا نفس المكتبات لكلا الشبكتين ,الفرق الوحيد هو هيكلية كل شبكة ,فمثلا شبكة CNN تختلف عن ANN في البنية ,فهي تملك طبقات مثل : Conv layer Max-pool Layer .

- **تحميل MNIST**

ثم يتم بعد الخطوة السابقة ,تحميل مجموعة البيانات التي سندرب عليها الشبكات ونختبرها, عن طريق تحميلها أو استدعائها مباشرة من الموقع الأصلي ,ويتم تقسيمها لمجموعتين ,مجموعة تدريب (X_train) و(Y_train) ,ومجموعة اختبار(X_test) و(Y_test) ,كما في الشكل(16):

```
(x_train,y_train),(x_test,y_test)=mnist.load_data()

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [==============================] - 0s 0us/step
11501568/11490434 [==============================] - 0s 0us/step
```

الشكل(16):تحميل MNIST

- المعالجة المسبقة (Pre-Processing)
- إعادة تشكيل البيانات (Reshape),كما الشكل(17):

```
x_train= x_train.reshape(x_train.shape[0],28,28,1)
x_test= x_test.reshape(x_test.shape[0],28,28,1)
```

الشكل(17):يوضح (Reshape)

إضافة بعد رابع لبيانات صور التدريب والاختبار, حيث يصبح شكل بيانات التدريب (60000,28,28,1), وشكل بيانات الاختبار(10000,28,28,1),والرقم (1),توضح أن الصور ذات تدرج رمادي حتى يتعرف عليها النموذج.

- تطبيع البيانات(Normalize)

نقوم بتحويل نوع البيانات من(Integer) الى (Float) ,ثم نقوم بقسمة كل قيمة بكسل على 255.

```
x_train =x_train.astype('float32')
x_test=x_test.astype('float32')
x_train/= 255.0
x_test/= 255.0
```

الشكل (18)تطبيع البيانات

- ترميز البيانات(One Hot encoder)

تحويل صيغة البيانات للمجموعات سواء مجموعة التدريب أو الاختبار من أعداد صحيحة (int) الى مصفوفة فئة ثنائية(binary class matrix),الشكل (19) يوضح الـ(one hot encoder)

0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

الشكل (19) ترميز البيانات

الشكل (20),يوضح الكود الخاص بالترميز ,وقد استخدمنا دالة (categorical)

```
n_class = 10
y_train = to_categorical(y_train,n_class)
y_test = to_categorical(y_test,n_class)
print(y_train[0])
```

```
[0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
```

الشكل(20)ترميز البيانات

5- مرحلة التطبيق

1.5 - بناء نموذج(Model) للشبكة ANN كما موضح في الشكل(21)

```
model = Sequential()
model.add(Flatten(input_shape=(28,28)))

model.add(Dense(256,activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(200, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(128 ,activation = 'relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))


model.add(Dense(10,activation='softmax'))
```

الشكل(21) بنية ANN

كما هو موضح بالشكل, يتكون النموذج أساسا من خمس طبقات أساسية بدءا من طبقةFlatten() ,بالإضافة الى الطبقات الاخرى.

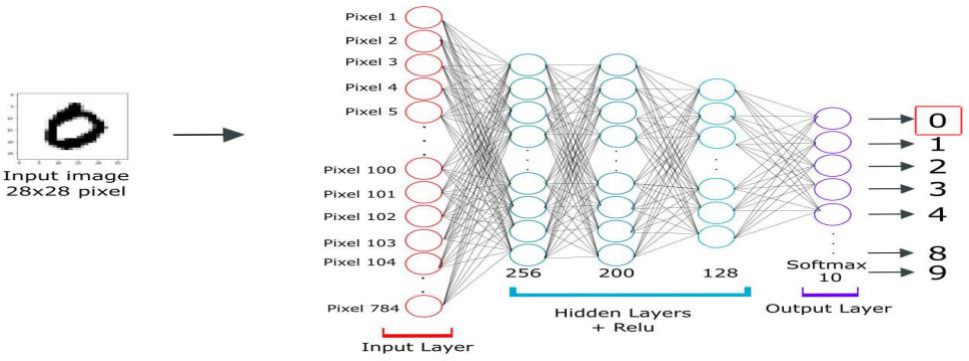
الطبقة الأولى ,تمثل طبقة الإدخال ,والتي تعتبر نواة النموذج.

الطبقة الثانية ,تعبر عن طبقة الاتصال بالكامل Dense() ,وقد ضمنا لها 256 عقدة (neuron) ,مع دالة تنشيط من نوع (Relu) ,وأتبعناها بطبقتين تحسين وهما,(Batch Normalization) ,و(Dropout) بنسبة إهمال 50 بالمئة.

والطبقة الثالثة, والرابعة نفس الطبقة السابقة ولكن واحدة بـ200عقدة ,و الاخرى بـ128 عقدة فقط

أما الطبقة الأخيرة فهي طبقة الإخراج ,والتي تتضمن عدد الفئات (Classes) ,وعددها 10 فئات بعدد الأرقام من 0 الى 9,مع دالة تنشيط من نوع (Softmax).

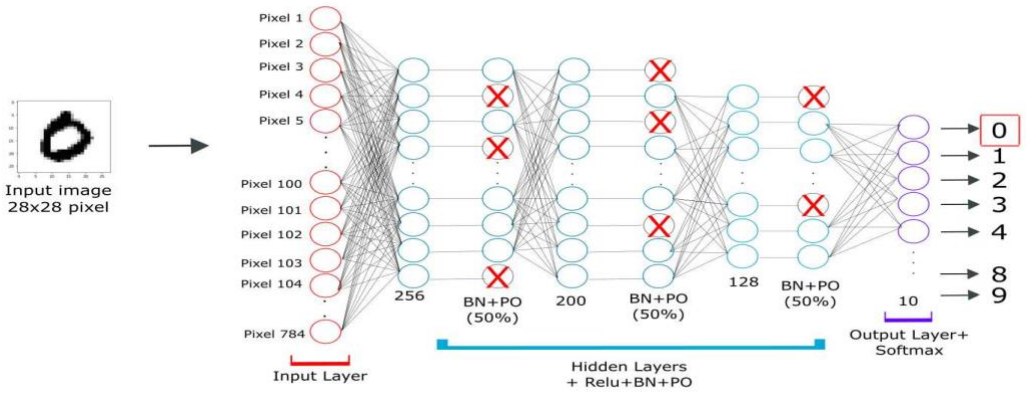
الشكل(22)يوضح هيكلية الشبكة بدون (Batch Normalization) و(Dropout)



الشكل (22) هيكلية ANN بدون (BN),(PO)

BN وPOهي تقنية مستخدمة على نطاق واسع في مجال التعلم العميق. إنه يحسن سرعة التعلم للشبكات العصبية ويوفر التنظيم وتجنب الإفراط في التجهيز.

والشكل(23)يوضح هيكلية الشبكة مع (Batch Normalization)و(Dropout)



الشكل(23)هيكلية ANN مع (BN), (PO)

3.5 بناء نموذج CNN,الشكل(24)يوضح بنية الشبكة

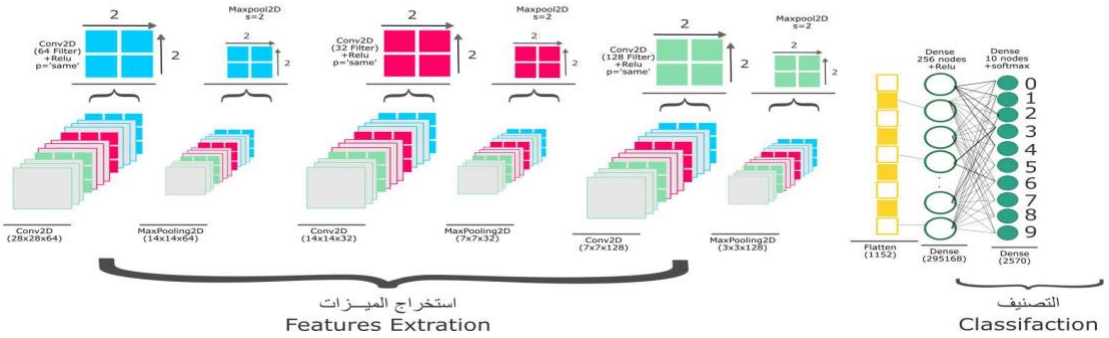
كما هو موضح الكود , يتكون نموذج CNN من ست طبقات أساسية, بالإضافة الى طبقات BN و PO .

```
model.add(Conv2D(64, kernel_size=2,padding = 'same',strides=1,activation ='relu', input_shape = (28, 28, 1)))
model.add(MaxPool2D(pool_size=(2, 2),strides=2))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Conv2D(32, kernel_size=2,strides=1,padding ='same', activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Conv2D(128,kernel_size=2, padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2),strides=2))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(n_class, activation='softmax'))
```

الشكل(24)بنية CNN

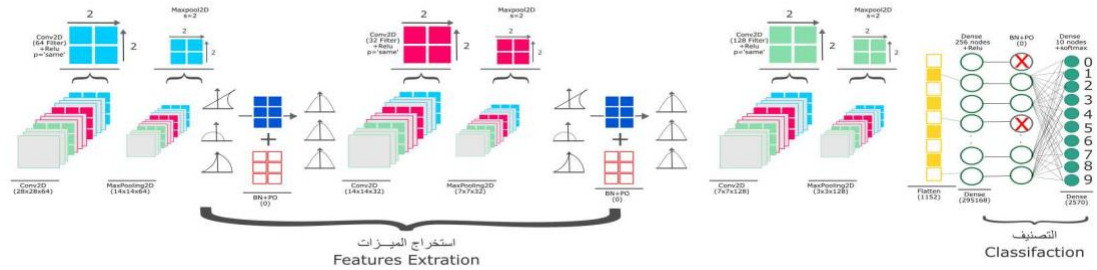
تمثل الطبقة الأولى طبقة الإدخال ,مكونة من طبقة التفاف بـــ64 مرشح (Filter),بأبعاد 3x3.بالإضافة الى الحشو نوع(Same) ,بخطوة واحدة, ودالة تنشيط من نوع (Relu) ,ويتبعها طبقة تجميع ذات أبعاد 2x2.مع العنصر الأساسي وهو Input_Shape=(28,28,1), والذي يمثل المدخل الذي يعتبر نواة النموذج.

في الطبقة الثانية تم استخدام 32 مرشح(Filter) بأبعاد 3x3.والطبقة الثالثة تم استخدام 128 مرشح بأبعاد3x3. في الطبقة الرابعة طبقة تسطيح لتحويل البيانات من مصفوفة الى متجه, وفي الطبقة الخامسة استخدمنا طبقة اتصال بالكامل بـــ256 عقدة ,مع دالة تنشيط(Relu).أما الطبقة الأخيرة فهي طبقة الإخراج ,وفيها عدد الفئات (n_class),بعدد الأرقام المراد التعرف عليها وهي 10,مع دالة تنشيط Softmax.الشكل(24)يوضح هيكلية CNN بدون BN, وPO .



الشكل(25) هيكلية CNN بدون BN ,وPO

و الشكل(26)يوضح هيكلية CNN مع BN ,و PO



الشكل(26)هيكلية CNN مع BN,وPO

4.5 تهيئة النمـــوذج (Model Compile)

هذه المرحلة تأتي قبل عملية التدريب, والغرض منها هو تهيئة(أو ترجمة)النموذج ,حتى نتمكن من تدريبه, الشفرة المكتوبة بالسفل توضح كود تهيئة النموذج

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

كما هو موضح في الشكل, فقد استخدمنا دالة الخسارة (categorical_crossentropy) ,لأنها تعتبر الأنسب في حالة التصنيف المتعدد(multiclass), كما استخدمنا خوارزمية التحسين (adam) ,وأخيرا عملية تتبع للدقة.

5.5 تدريب النموذج (Fit Model)

تحديد البيانات التي نريد أن ندرب الشبكة عليها, وهي(x_train,y_train), ومن ثم تحديد حجم الدفعة وهي = 35 ,واختيار فترات التدريب وهنا اخترنا فترات مختلفة, ونحدد بيانات التحقق من الصحة بنسبة 30 بالمئة . الشفرة المكتوبة بالسفل توضح كود عملية التدريب النموذج

```
hist =model.fit(x_train,y_train, validation_split=0.3,epochs=100, batch_size=35,verbose=1)
```

6- مرحـلة الاختبــار و النتائج

1.6 اختبار النموذج (Test Model)

في هذه المرحلة نستدعي النموذج الذي تم حفظه, ونقوم بالتعرف على أرقام مجموعة التدريب من خلال رقم الفهرس.

1.1.6 التعرف على أرقام جديدة

الشكل (27),يوضح لنا المعالجة المسبقة للصور المدخلة ,وذلك بأول خطوة نقوم بها ,وهي استدعاء مكتبات معالجة الصور(preprocessing),وتتمثل الخطوات التي بعدها ,بجعل الصور رمادية, وتحجيمها بحجم معين(28,28),ثم تحويلها من (int) الى (float) ,واخيرا قسمتها على 255 حتى تكون في نطاق محدد.

```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.models import load_model


def load_image(filename):
 img = load_img(filename, grayscale=True, target_size=(28,28))
 img = img_to_array(img)
 img = img.reshape(1,28,28)
 img = img.astype('float32')
 img = img / 255
 return img
```

الشكل(27)يوضح معالجة الصور

2.6 النتائج و الاستنتاجات

في هذا الجزء نستعرض نتائج كل من ANN وCNN ومقارنتها بناء على الدقة والأداء والوقت ,بالإضافة الى تغيير عدد الفترات , كذلك مقارنة مدى فعاليتها باستخدام طبقات BN+PO وبدون استخدام طبقات BN+PO.

1.2.6 النتائج بدون (BN+PO)

الان الجداول التالية توضح الفروقات في حال إضافة BN+PO

- عدد الفترات (Epochs= 5,10,25)
- حجم الحزمة (Batch size = 64)

الجدول(1) نتائج ANN

الأرقام	عدد الفتـرات(Epochs)		
	5	10	25
0	%66 – 5	%37 – 4	%24 – 6
1	%49 – 5	%54 – 1	%61 – 8

2	%21 – 8	%42 – 2	%92 – 2
3	%37 – 1	%58 – 1	%42 – 1
4	%72 – 4	%88 – 4	%79 – 4
5	%60 – 5	%61 – 4	%99 – 5
6	%56 – 5	%57 – 4	%50 – 5
7	%44 – 1	%93 – 1	%61 – 7
8	%45 – 4	%68 – 4	%54 – 4
9	%44 – 4	%72 – 1	%42 – 7
الدقة	%89	%92	%94
الوقت المنجز	33 ث	1 د 43 ث	1 د 44 ث

الجدول(2)نتائج CNN

الأرقام	**عدد الفتـرات(Epochs)**		
	5	10	25
0	%39 – 1	%38 – 1	%98 – 1
1	%99 – 1	%99 – 1	%99 – 1
2	%49 – 1	%60 – 3	%49 – 1
3	%96 – 1	%85 – 3	%42 – 1
4	%63 – 4	%71 – 1	%79 – 4
5	%54 – 5	%93 – 5	%99 – 5
6	%53 – 9	%34 – 5	%50 – 5
7	%65 – 7	%96 – 3	%61 – 7
8	%53 – 1	%58 – 5	%54 – 4
9	%82 – 9	%28 – 1	%42 – 7
الدقة	%91	%94	%96
الوقت المنجز	7 د 23 ث	12 د 22 ث	24 د

نلاحظ ان نتائج CNN لازلت افضل من ANN

- الان في حالت استخدام حجم الحزمة 35

الجدول(3)نتائج ANN

الأرقام	عدد الفتــرات(Epochs)		
	5	10	25
0	%34 – 4	%46 – 4	%71 – 4
1	%58 – 1	%40 – 6	%59 – 8
2	%82 – 2	%77 – 2	%73 – 2
3	%85 – 1	%67 – 1	%63 – 1
4	%88 – 4	%78 – 4	%90 – 4
5	%60 – 5	%63 – 6	%66 – 4
6	%33 – 5	%23 – 5	%75 – 4
7	%66 – 1	%63 – 7	%48 – 7
8	%33 – 1	%61 – 1	%44 – 4
9	%34 – 1	%31 – 7	%29 – 1
الدقة	%89	%91	%94
الوقت المنجز	48 ث	1د 30 ث	4 د 23 ث

CNN الجدول(4)نتائج

الأرقام	عدد الفتــرات(Epochs)		
	5	10	25
0	%95 – 1	%74 – 0	%46 – 3
1	%99 – 1	%99 – 1	%99 – 1
2	%37 – 3	%69 – 2	%37 – 5
3	%89 – 1	%88 – 3	%61 – 3
4	%96 – 1	%77 – 4	%46 – 4

5	%85 – 5	%30 – 5	%77 – 5
6	%48 – 1	%74 – 9	%52 – 1
7	%64 – 3	%63 – 7	%59 – 3
8	%57 – 9	%64 – 9	%45 – 1
9	%88 – 1	%56 – 9	%33 – 9
الدقة	%90	%94	%96
الوقت المنجز	7 د 23 ث	13 د 22 ث	31 د 24 ث

مع تغير حجم الحزمة الى 35 نقصت دقت CNN قليلا

- هذه الاختبار مع حجم الحزمة 20

الجدول(5)نتائج ANN

الأرقام	**عدد الفتـرات(Epochs)**		
	5	10	25
0	%51 – 4	%68 – 4	%40 – 4
1	%44 – 1	%41 – 1	%83 – 1
2	%45 – 2	%85 – 2	%82 – 2
3	%63 – 1	%74 – 1	%68 – 1
4	%84 – 4	%93 – 4	%79 – 4
5	%52 – 5	%57 –5	%78 – 5
6	%37 – 4	%31 – 6	%64 – 5
7	%97 – 1	%82 – 1	%36 – 7
8	%33 – 4	%39 – 5	%69 – 4
9	%48 – 1	%29 – 2	%37 – 4
الدقة	%87	%90	%93
الوقت المنجز	1د 23 ث	3 د 24 ث	6 د 23 ث

الجدول(6)نتائج CNN

الأرقام	عدد الفتــرات(Epochs)		
	5	10	25
0	%43 – 3	%96 – 1	%69 – 1
1	%99 – 1	%99 – 1	%99 – 1
2	%28 – 3	%34 –1	%69 – 1
3	%58 – 1	%88 – 1	%98 – 1
4	%95 – 1	%51 – 4	%38 – 1
5	%88 – 5	%78 –5	%64 – 5
6	%50 – 5	%70 – 1	%84 – 1
7	%50 –3	%70 – 3	%75 – 7
8	%44 – 5	%65 – 9	%62 –1
9	%68 – 9	%47 – 9	%58 – 9
الدقة	%91	%94	%95
الوقت المنجز	7 د 23 ث	15 د 23 ث	36 د 33 ث

1.2.6 .النتائج مع (BN+PO)

الجداول التالية توضح الفروقات مع استخدام BN+PO ومع الشروط التالية:

- ✓ عدد الفترات (Epochs = 5,10 ,25)
- ✓ حجم الحزمة (Batch size =64)

الجدول(7) نتائج ANN

الأرقام	عدد الفتــرات(Epochs)		
	5	10	25
0	%45 - 8	%35 - 5	%45 – 5
1	%81 - 8	%91 - 8	%72 – 8
2	%69 - 2	%36 - 5	%58 – 2

3	%56 - 1	%48 - 1	%42 – 2
4	%54 - 7	%36 - 2	%39 – 7
5	%59 - 5	%88 - 5	%85 – 5
6	%35 - 5	%81 – 5	%73 – 8
7	%53 - 7	%53 – 3	%53 – 1
8	%39 - 1	%70 – 9	%45 – 4
9	%54 - 8	%37 – 3	%61 – 3
الدقة	96	97	98
الوقت المنجز	24 ث	1د و 22ث	2 د

نلاحظ ان في جدول (7) كلما زادة عدد الفترة زادت الدقة. وكذلك الحال في جدول (8) مع تقنية CNN.

الجدول (8)نتائج CNN

الأرقام	**عدد الفتــرات(Epochs)**		
	5	10	25
0	%51 – 3	%45 – 3	%35 – 0
1	%77 – 1	%98 – 1	%56 – 1
2	%43 – 3	%67 – 1	%92 – 5
3	%61 – 3	%64 - 1	%89 – 3
4	%72 – 4	%91 – 4	%47 – 7
5	%74 – 3	%43 – 3	%94 – 5
6	%27 – 7	%40 – 2	%68 – 5
7	%56 - 7	%57 – 3	%53 – 3
8	%63 – 9	%85 – 9	%73 – 5
9	%69 – 7	%95 – 9	%94 – 9
الدقة	%97	%98	%99
الوقت المنجز	5 د	9 د 22 ث	22 د

- في هذه التجربة قم بتغير حجم الحزمة (Batch size = 35)

الجدول(9) نتائج ANN

الأرقام	عدد الفتـرات(Epochs)		
	5	10	25
0	%50 – 5	%45 – 6	%11 – 0
1	%35 – 6	%99 – 8	%11 – 0
2	%36 – 2	%91 – 2	%11 – 0
3	%72 – 1	%41 – 1	%12 – 7
4	%25 – 3	%98 – 3	%11 – 0
5	%82 – 5	%83 – 5	%11 – 0
6	%36 - 5	%39 – 6	%12 – 0
7	% 29 – 1	%50 – 7	%12 – 0
8	%23 - 1	%71 – 4	%12 – 0
9	%22 – 7	% 56 – 3	%11 – 0
الدقة	%96	%98	%98
الوقت المنجز	3 د 21 ث	1د و 23ث	2 د

الجدول(10) نتائج CNN

الأرقام	عدد الفتـرات(Epochs)		
	5	10	25
0	%43 – 3	%43 – 7	%70 – 7
1	%84 – 1	%62 – 1	%69 – 1
2	%55 – 3	%48 – 7	%79 – 3
3	%68 – 1	%46 – 2	%83 – 3
4	%66 – 4	%98 – 4	%72 – 4
5	%50 – 3	%59 – 5	%89 – 9
6	%26 - 5	%33 – 3	%55 – 1
7	%82 – 3	%89 – 7	%55 – 3

8	%52 – 5	%63 – 9	%64 – 9
9	%34 – 1	%78 – 9	%73 – 7
الدقة	%97	%98	% 99
الوقت المنجز	5 د 22 ث	10د و 19ث	25 د 21 ث

نلاحظ عند تغير حجم الحزمة الى 35 Batch Size لا يوجد الفرق كبير بين النتائج.

-و في هذا الاختبار قم بتغير حجم الحزمة (Batch size =20)

الجدول(11) نتائج ANN

الأرقام	**عدد الفتـرات(Epochs)**		
	5	10	25
0	%33 – 8	%65 – 5	%34 – 8
1	%66 – 8	%41 – 6	%63 – 1
2	%98 – 2	%77 – 2	%99 – 2
3	%29 – 2	%48 – 6	%57 – 6
4	%40 – 4	%30 – 2	%40 – 2
5	%78 – 5	%97 – 5	%79 – 5
6	%38 – 5	%34 – 5	%31 – 5
7	%38 – 7	%42 – 1	%88 – 1
8	%32 – 9	%85 – 4	%85 – 1
9	%38 – 9	%48 – 3	%39 – 3
الدقة	%99	%97	%99
الوقت المنجز	52 ث	1 د 48 ث	5 د 22 ث

الجدول(12) نتائج CNN

الأرقام	**عدد الفتـرات(Epochs)**		
	5	10	25

0	%45 – 5	%22 – 8	%96 – 3
1	%99 – 1	%96 – 1	%99 – 1
2	%93 – 5	%89 – 2	%99 – 3
3	%44 – 2	%33 – 1	%80 – 1
4	%66 – 1	%95 – 4	%99 – 4
5	%51 – 9	%63 – 9	%79 – 9
6	%65 – 5	%19 – 2	%73 – 3
7	%78 – 3	%69 – 2	%87 – 3
8	%83 – 9	%71 – 9	%77 – 5
9	%65 – 4	%97 – 9	%51 – 9
الدقة	%98	%98	%99
الوقت المنجز	6 د 23 ث	11 د 22 ث	22 د

تم استاج من هذا الاختبار مع استخدام BN+PO ومع تغير حجم الحزمة لا يوجد تغير ملحوظ ولكن بشكل عام تقنية CNN افضل نتائج من ANN.

2.4 الاستنتاجات

- ✓ الدقــة: لحضنا CNN هي الافضل في الدقة وصلت الى 99% مع عد الفترة 25.
- ✓ الأداء : من الجداول الموضحة أعلاه ,تبين أن CNN استطاعت أن تتعرف على اغلب الارقام من أصل 10 أرقام ,بينما ANN استطاعت أن تتعرف على حوالي نصف الارقام
- ✓ الوقت : بالنسبة لوقت التدريب فتعتبر ANN أسرع من CNN
- ✓ عدم التعلم : تعتبر ANN أكثر عرضة لفرط التجهيز ,وCNN أقل عرضة منها.
- ✓ تبين كذلك أن طبقات التحسين BN+PO,استطاعت أن تجنبنا فرط التجهيز بنسبة جيدة ,كما استطاعت من تحسين الشبكات للتعلم.

7- الخاتمة

تم بحمد الله إتمام العمل, ,وقد أدى لنتائج مقبولة لغرض المقارنة ,وتوضحت لدينا أداء كل شبكة وما يميز كل شبكة عن الأخرى. مما تبين لن ان تقنية CNN استطاعة تميز الارقام المكتوبة بخط اليد افضل من تقنية ANN بجولي الضعف.كذلك ساعدت طبقات BN+PO في تحسين أداء الشبكة .ومن ناحة الزمن فان ANN هي الافضل. و لتطوير هذا الدراسة وتحسين أدائه في المستقبل, ننصح عمل الاتي:

- استخدام قاعدة بيانات اكبر وهي التي تعطي للبرنامج مجال علم اكبر للتعلم واعطاء نتائج افضل جديدة للشبكات ,يمكن أن يدرب الشبكة ,يجعلها تتعلم اكثر
- التقليل من حجم الدفعات يجعل من الشبكة تتعلم بشكل صحيح
- استخدام طبقات تحسين بما يتناسب مع حجم الشبكات (عدد الطبقات)
- توفير وقت كافي للشبكات حتى تتعلم ,وذلك بزيادة عدد الفترات

المـراجـع

1. Jagtap, V. N., & Mishra, S. K. (2014). Fast efficient artificial neural network for handwritten digit recognition. *International Journal of Computer Science and Information Technologies*, *5*(2), 2302-2306.
2. Saeed, A. M. (2015). Intelligent handwritten digit recognition using artificial neural network. *Int. Journal of Engineering Research and Applications*, *5*(5), 46-51.
3. Sakshica, D., & Gupta, K. (2015). Handwritten digit recognition using various neural network approaches. *Int. J. Adv. Res. Comput. Commun. Eng*, *4*(2).
4. Roshni Musaddi.2018 . Handwritten Digit Recognition using Neural Network. Journal of Network Communications and Emerging Technologies (JNCET)

.Volume 8, Issue 10. ISSN: 2395-5317

5. Hamid, N. A., & Sjarif, N. N. A. (2017). Handwritten recognition using SVM, KNN and neural network. arXiv preprint arXiv:1702.00723.

6. Abdulrazzaq, M. B., & Saeed, J. N. (2019, April). A comparison of three classification algorithms for handwritten digit recognition. In *2019 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 58-63). IEEE.
7. Tabik, S., Peralta, D., Herrera-Poyatos, A., & Herrera Triguero, F. (2017). A snapshot of image pre-processing for convolutional neural networks: case study of MNIST.

8. Ghosh, M. M. A., & Maghari, A. Y. (2017, October). A comparative study on handwriting digit recognition using neural networks. In 2017 international conference on promising electronic technologies (ICPET) (pp. 77-81). IEEE.

9. Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved handwritten digit recognition using convolutional neural networks (CNN). *Sensors*, *20*(12), 3344.

10. Vinjit, B. M., Bhojak, M. K., Kumar, S., & Nikam, G. Implementation of Handwritten Digit Recognizer using CNN.
11. Pashine, S., Dixit, R., & Kushwah, R. (2021). Handwritten Digit Recognition using Machine and Deep Learning Algorithms. *arXiv preprint arXiv:2106.12614*.
12. Alwzwazy, H. A., Albehadili, H. M., Alwan, Y. S., & Islam, N. E. (2016). Handwritten digit recognition using convolutional neural networks. *International Journal of Innovative Research in Computer and Communication Engineering*, *4*(2), 1101-1106.
13. Vaidya, S. A., & Bombade, B. R. (2013). A novel approach of handwritten character recognition using positional feature extraction. *International Journal of Computer Science and Mobile Computing*, *2*(6), 179-186.
14. Yadav, P., & Yadav, N. (2015). Handwriting recognition system-a review. *International Journal of Computer Applications*, *114*(19), 36-40.
15. Neural Networks and Deep Learning, By Michael Nielsen / Dec 2019

    /http://neuralnetworksanddeeplearnin